
Data Structures and Algorithms

COMP 103

2019-20

Semester 2

Lecture 10b

Dr. Kerese Manueli

kerese.manueli@ecs.vuw.ac.nz

Victoria University of Wellington

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

If you aren't familiar with the game, you can run the demo, or play the KDE version (KMines - in the Games menu, under Tactics and Strategy).

The `Minesweeper` class has code for most of the program, except for the key methods of

- `mark(row, col)`: The player has clicked on a square to mark it.
- ✓ `tryExpose(row, col)`: the player has clicked on a square to expose it
- ✓ `exposeSquareAt(row, col)`: expose the square at (row,col) and if it has no adjacent mines, spread the exposing to all connected safe squares. This method should be recursive, and needs to spread out horizontally, vertically, and diagonally.
- `hasWon()`: check whether the player has won the game yet (exposed all the squares without a mine).

The `Square` class has code for individual squares; you need to use it, but you should not modify it.

Core and Completion:

Complete the four incomplete methods in `Minesweeper` in order to make the game work.

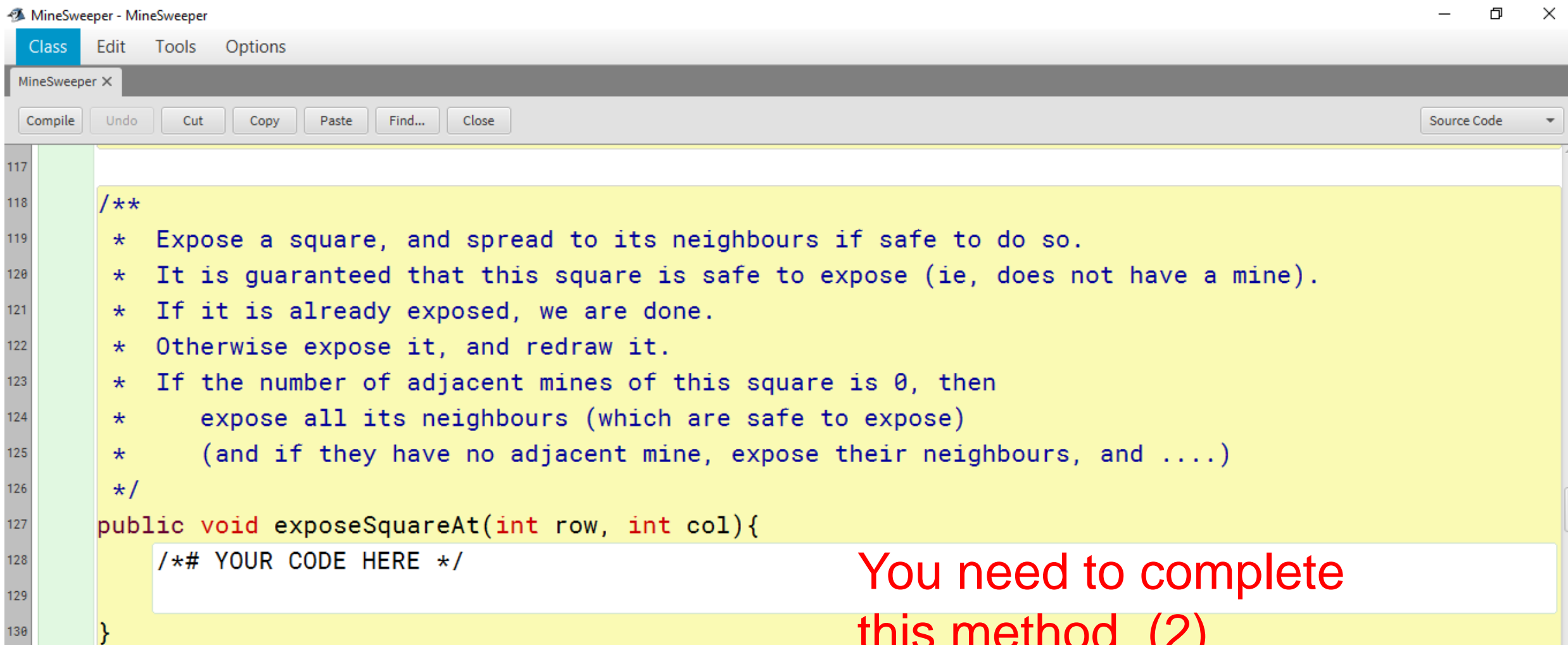
Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
101
102 /**
103  * The player has clicked on a square to expose it
104  * - if it is already exposed or marked, do nothing.
105  * - if it's a mine: lose (call drawLose())
106  * - otherwise expose it (call exposeSquareAt)
107  * then check to see if the player has won and call drawWon() if they have.
108  * (This method is not recursive)
109  */
110 public void tryExpose(int row, int col){
111     /*# YOUR CODE HERE */
112
113     if (hasWon()){
114         drawWin();
115     }
116 }
```

You need to complete this method (1)

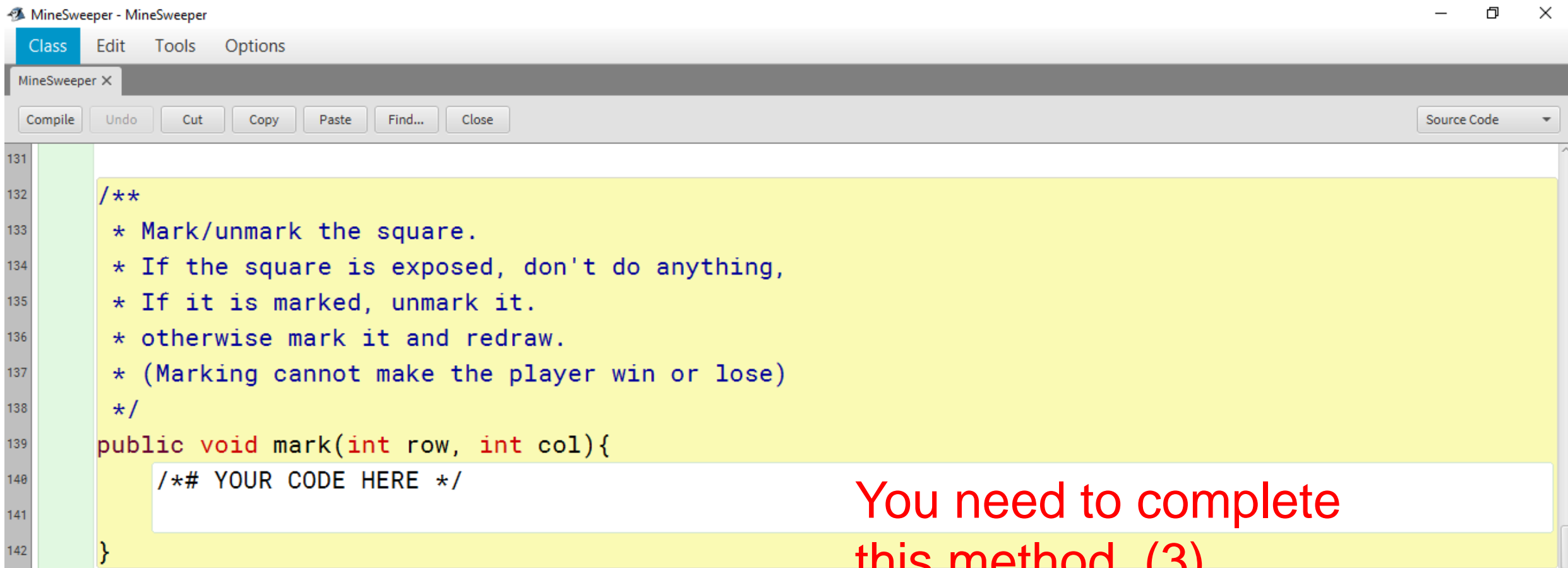
Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
117
118 /**
119  * Expose a square, and spread to its neighbours if safe to do so.
120  * It is guaranteed that this square is safe to expose (ie, does not have a mine).
121  * If it is already exposed, we are done.
122  * Otherwise expose it, and redraw it.
123  * If the number of adjacent mines of this square is 0, then
124  *     expose all its neighbours (which are safe to expose)
125  *     (and if they have no adjacent mine, expose their neighbours, and ....)
126  */
127 public void exposeSquareAt(int row, int col){
128     /*# YOUR CODE HERE */
129
130 }
```

You need to complete
this method (2)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
131
132  /**
133   * Mark/unmark the square.
134   * If the square is exposed, don't do anything,
135   * If it is marked, unmark it.
136   * otherwise mark it and redraw.
137   * (Marking cannot make the player win or lose)
138   */
139  public void mark(int row, int col){
140      /*# YOUR CODE HERE */
141  }
142
```

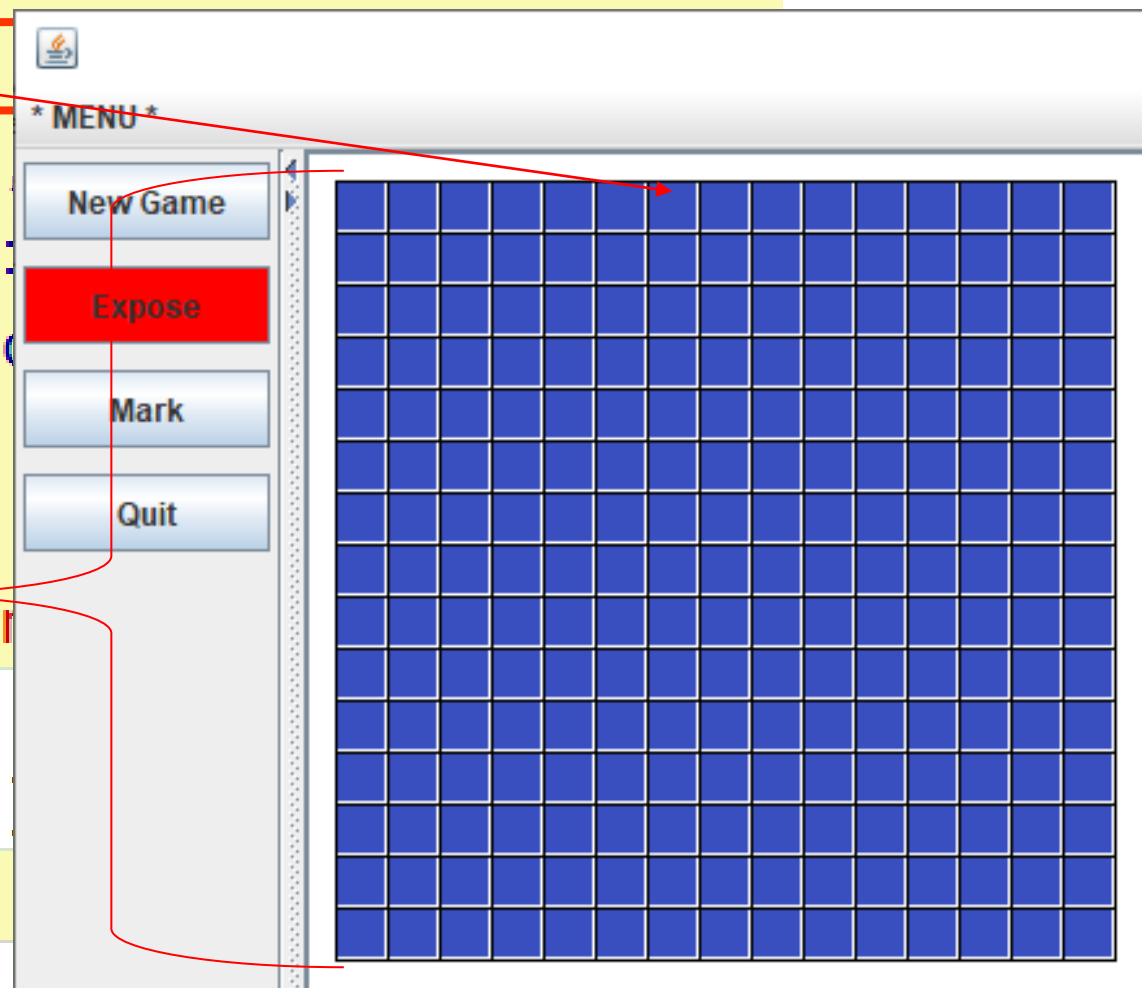
You need to complete
this method (3)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154  * Mark/unmark the square.
155  * If the square is exposed
156  * If it is marked, unmark
157  * otherwise mark it and rec
```

```
32 public class Minesweeper {
33
34     public static final int ROWS = 15;
35     public static final int COLS = 15;
    // YOUR CODE HERE
    Square sq = squares[row
162
163 }
```



Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154  * Mark/unmark the square.
155  * If the square is exposed, don't do anything,
156  * If it is marked, unmark it.
157  * otherwise mark it and redraw.
158  * (Marking cannot make the player win or lose)
159  */
160  public void mark(int row, int col){
161      /*# YOUR CODE HERE */
162      Square sq = squares[row][col];
163  }
```

(1)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154  * Mark/unmark the square.
155  * If the square is exposed, don't do anything,
156  * If it is marked, unmark it.
157  * otherwise mark it and redraw.
158  * (Marking cannot make the player win or lose)
159  */
160  public void mark(int row, int col){
161      /*# YOUR CODE HERE */
162      Square sq = squares[row][col];
163  }
```

(2)

```
if (sq.isExposed())
    return;
```


Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154     * Mark/unmark the square.
155     * If the square is exposed, don't do anything,
156     * If it is marked, unmark it.
157     * otherwise mark it and redraw.
```

toggleMark() loaded in the Square class

```
159
160 80  /** Toggle the mark */
161 81  public void toggleMark(){
162 82     this.marked = ! this.marked;
163 83  }
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154   * Mark/unmark the square.
155   * If the square is exposed, don't do anything,
156   * If it is marked, unmark it.
157   * otherwise mark it and redraw.
158   * (Marking cannot make the player win or lose)
159   */
160  public void mark(int row, int col){
161      /*# YOUR CODE HERE */
162      Square sq = squares[row][col];
163  }
```

(3) `sq.toggleMark();`

Assignment 3b

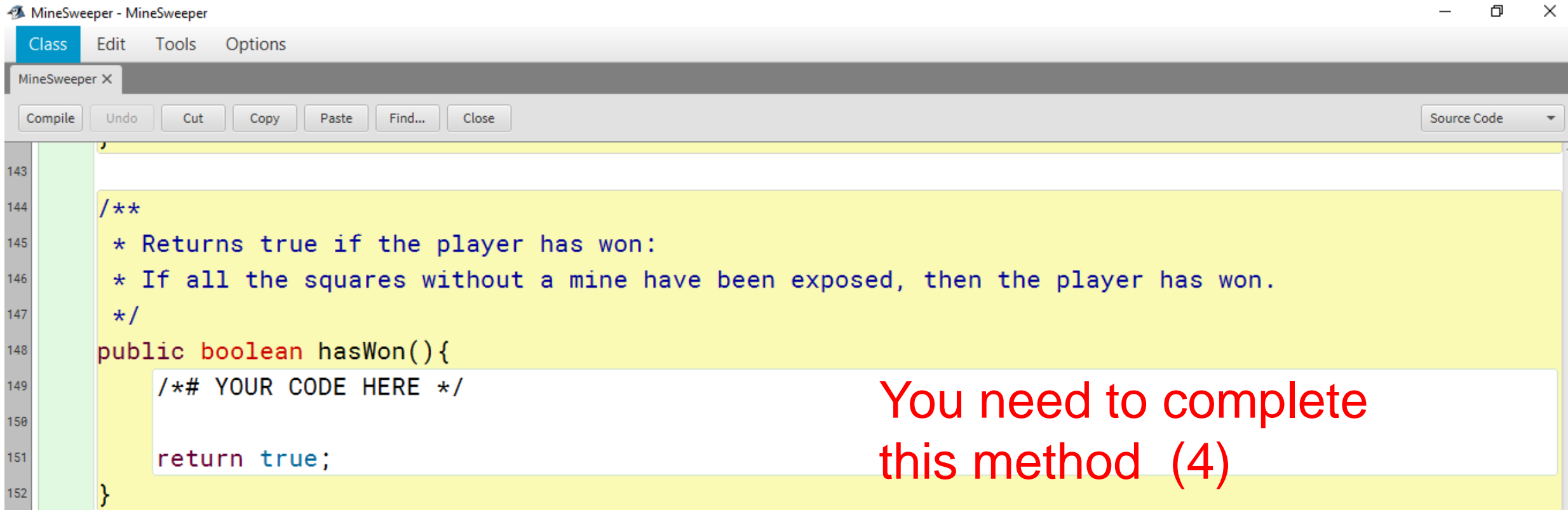
https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
153  /**
154   * Mark/unmark the square.
155   * If the square is exposed, don't do anything,
156   * If it is marked, unmark it.
157   * otherwise mark it and redraw.
158   * (Marking cannot make the player win or lose)
159   */
160  public void mark(int row, int col){
161      /*# YOUR CODE HERE */
162      Square sq = squares[row][col];
163  }
```

(4)

```
sq.draw(LEFT+col*SQUARE_SIZE, TOP+row*SQUARE_SIZE, SQUARE_SIZE);
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
143
144 /**
145  * Returns true if the player has won:
146  * If all the squares without a mine have been exposed, then the player has won.
147  */
148 public boolean hasWon(){
149     /*# YOUR CODE HERE */
150
151     return true;
152 }
```

You need to complete this method (4)

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB

```
* Returns true if the player has won:  
* If all the squares without a mine have been exposed,  
* then the player has won.  
*/  
public boolean hasWon(){  
    /*# YOUR CODE HERE */  
    for  
    {  
        for  
        {  
        }  
    }  
    return true;  
}
```

Assignment 3b

https://ecs.wgtn.ac.nz/Courses/XMUT103_2020T1/Assignment3PartB